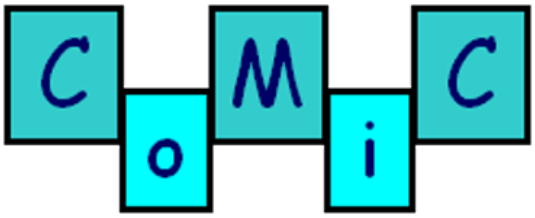
Author: Michael White Date: 24 September 2004

Deliverable 7.4
Experiments with Multimodal Output in Human-Machine Interaction



Document history

Date	Editor	Explanation	Status
6 September	MWW	First draft	Draft
24 September	MWW	Comments from Norbert and Louis tB	Final

COMIC

Information sheet issued with Deliverable 7.4

Title:	Experiments with Multimodal Output in Human-Machine Interaction
Abstract:	We describe the experiments with multimodal output undertaken in the COMIC project to date, emphasizing instance-based English surface realization and its connection with speech synthesis. In particular, we show (1) how n-grams from a training corpus can be used to guide the realizer to quickly produce target realizations with high accuracy, and (2) that the resulting realizations positively affect the perceived quality of synthesized speech produced from them, when compared to baseline realizations, with very high significance.
Author(s):	Michael White
Reviewers:	Louis ten Bosch, Norbert Pfleger
Project:	COMIC
Project number:	IST-2001-32311
Date:	24 September 2004
	Public
Key words:	Surface realization, text-to-speech synthesis, unit selection, intonation
Distribution list	
COMIC partners:	Edinburgh, DFKI, KUN, MPI-N, MPI-T, Sheffield, ViSoft
External COMIC:	PO, reviewers

Contents

1 Introduction

This report described the experiments with multimodal output undertaken in the COMIC project to date. The emphasis of the report is on English surface realization and its connection with speech synthesis, as these are the two aspects of multimodal output that have not been examined in other deliverables. In particular, we show how a corpus of training examples can be used to guide decision making in our hybrid symbolic-statistical surface realizer, and how the resulting realizations—annotated with prosodic output—positively affect the perceived quality of synthesized speech produced from these realizations.

The report is organized as follows. In section 2, we briefly review the related experiments on multimodal output in COMIC. In section 3, we report on unpublished work showing how n-grams derived from a corpus of target examples can be used to guide anytime chart realization. In section 4, we report the results of a perception experiment to appear in Rocha's forthcoming MSc thesis [15], showing that preferred realizations, when synthesized, are perceived as better than baseline ones. Finally, in section 5, we conclude with a discussion of future directions.

2 Related Experiments

2.1 Face

Deliverable 1.3 examines the usability of the Year 2 version of the COMIC system, and includes the results of an experiment comparing the impact of facial expressions on the conversational interaction in Phase 3. In this study, we found that the thinking expression displayed at the end of the user's turn helped to convey that the system was busy processing input, and that the subsequent nods, smiles or confused expressions provided an early visual indication of the system's success in processing user input, thereby mitigating the system's perceived sluggishness in responding verbally. However, we also found that the facial expressions face had a negative impact on task success and ease; this may have been because the expressive face to some extent distracted subjects from the task of examining the different tiling possibilities—in comparison to a version of the face with the expressions turned off—without improving the interaction enough to compensate for the distraction. Finally, while the user comments indicated a trend towards finding the expressive face more natural, we found no significant difference in general liking or overall satisfaction. This could be because the positive and negative effects of the expressive face cancelled each other out, and/or that any effect of the face condition was swamped by the overall clumsiness of the multimodal interaction in this preliminary version of the system.

2.2 Gestures

Deliverable 6.5 describes a study designed to compare user evaluations of the mouse-pointer gestures planned by two versions of the T24 fission module, one using rule-based gestures, and the other stochastically generated gestures based on data from an earlier corpus study. The experiment showed that subjects were generally able to tell the difference between the gestures planned by the two modules, and largely preferred those generated by the rule-based module to those from the stochastic module. The main reasons subjects gave for their preferences were that the rule-based versions were better synchronised with the speech, and that there were always gestures at the thumbnails during the second half of the description. Closer inspection of the stochastic versions revealed that while these were sometimes perceived as more natural, there were several problematic aspects to them that could perhaps be improved by making the stochastic decisions in a coordinated way, rather than all independently. In the next version of the fission module, we will therefore experiment with making some choices in a rule-based way, for consistency, while deferring others to the surface realizer, taking advantage of the planned extensions to its n-gram scoring mechanism in order to make any remaining gesture choices in a coordinated, example-based fashion.

2.3 Voice

At MPI Nijmegen, Aoju Chen is currently investigating how pitch accent type influences the interpretation of information status. The results of this study will be appropriate for publication in international journals such as *Speech Communication*. A brief description of the motivation, hypotheses and method of her experiment follows.

It is well established that in languages such as English, placement of pitch accent (i.e. accentuation vs. deaccentuation) is of great importance in the interpretation of information status. An ill-understood area is the role of pitch accent type in this respect. According to discourse-oriented models of intonational meaning in British English, given information can also be signalled by accentuation, just like new information; crucial is the choice of pitch accent. That is, some pitch accents would seem to connote the notion of 'new' information (hereafter 'new' accents), whereas other pitch accents the notion of 'given' information (hereafter 'given' accents). Taking these models as a starting point, this study will examine the interpretation of information status under conditions where the word in question is said with different pitch accents in British English.

On the basis that accentuation tends to mark a discourse entity as new and deaccentuation as given in a discourse, we arrive at the following hypotheses:

Hypothesis 1 The postulated 'new' accents will lead listeners to interpret the discourse entity at issue as previously unmentioned but the postulated 'given' accents will, like deaccentuation, lead listeners to interpret the discourse entity as previously mentioned.

Hypothesis 2 : These biases in listeners' interpretation mentioned in Hypothesis 1 will be intensified when the discourse entity in question is said with intensified 'new' accents and intensified 'given' accents.

The hypotheses will be tested by means of the eye-tracking paradigm. In the experiment, subjects will follow instructions (e.g. *Put the candle/candy below the triangle; now put the candle above the square*) recorded in a human voice as well as synthesized in the Festival system, and move objects displayed on a computer screen by the help of computer mouse. Subjects' eye fixations on pictured entities will be monitored while they are performing the task. In each trial, two of the objects will share the same stressed syllable (e.g. *candle* vs. *candy*) such that the target noun (i.e. *candle*) in the second part of the instruction is temporarily ambiguous during the first syllable and both the target noun and the competitor (i.e. *candy*) are potential referents at that stage. The target word in the second part of the instruction will be realised with two 'new' accents and two 'given' accents as well as without accent. It is predicted that the effect of pitch accent type will be reflected in the proportion of fixations to the competitor, and that this bias in listeners' eye fixation patterns will be intensified when the target word is said with the intensified accent.

3 Using N-grams to Guide Anytime Chart Realization

3.1 Overview

For English output, we have been developing a new general purpose, open source surface realizer, OpenCCG,[1] based on Combinatory Categorial Grammar. The T24 system marks the first occasion in which the new realizer is used in COMIC.

Using a state-of-the-art grammar formalism such as CCG in surface realization makes it possible to achieve more natural and varied output than is possible with simpler methods—which, unless carefully crafted, tend to be either repetitive or ungrammatical—but poses significant challenges for both efficiency and knowledge acquisition. In particular, the naïve chart realization algorithm at the core of our approach is exponential in worst-case complexity, so various techniques for reducing the search space are needed for it to be reasonably efficient in practice; at the same time, the more flexible the realizer becomes in its output potential, the greater the knowledge acquisition burden typically becomes for acquiring rules to choose forms appropriate to the context.

[1] `http://openccg.sourceforge.net/`

To help address both of these challenges, we have been investigating novel techniques for instance-based scoring of alternative (partial) realizations. Generally speaking, instance-based generation involves a memory-based approach to making generation decisions, where a set of target instances of outputs that are known to be good are compared against the alternatives under consideration by the generator. In instance-based realization, the instance base may be consulted to help choose among different possible word orders, syntactic constructions, or even among alternative lexical realizations of the input predicates—and their potential combinations. The instance-based approach alleviates the knowledge acquisition burden by reducing the need to acquire explicit rules to make such realization decisions, which may involve subtle interactions that are difficult to model. However, in order to be effective, care must be taken to establish an appropriate instance base, and methods must be found to generalize from the specific targets in the instance base (whose semantics rarely match the current input exactly).

To make the realizer fast enough for use in COMIC, we have been investigating a combination of novel techniques, including: (i) using rules to chunk the input logical form into sub-problems to be solved independently prior to further combination; (ii) pruning edges from the chart (with equivalent categories) based on the n-gram score of the (partial) string; and (iii) formulating the search as an anytime algorithm that can return the best available realization (according to its n-gram score) at the end of a fixed time period. Together, these techniques succeed in making the realizer fast enough for the T24 system; without them, it would be quite sluggish on average, and way too slow on the more difficult cases.

A specific technical challenge for T24 has been to implement a grammar and lexicon that covers the desired range of outputs, including the specifications of such prosodic elements as pitch accents, boundary tones and phrase breaks, as well as the desired links between words and both deictic gestures and face commands. The prosodic elements in the output are chosen to realise elements of information structure in the input, subject to the constraints imposed by the grammar on the mapping between information structure and prosody, following CCG theory. The n-gram scoring mechanism has also been adapted to handle the prosodic elements in the outputs, so that rule- and data-driven techniques may be combined in making the specific prosodic choices in the best available realization.

The output of the realizer is a simplified subset of APML (Affective Presentation Markup Language) [7], containing elements for pitch accents and boundary tones, plus labelled spans corresponding to the marked nodes in the input logical form (LF). The output can be trivially converted to APML by the synthesiser module, for input to Festival. Festival currently accepts prosodic specifications in APML with its diphone voices, and work is in progress on APML-enabling unit selection voices.

For details of the algorithm and a case study comparing the various efficiency methods, see [18, 19]. In the rest of this section, we present unpublished experiments showing how the amount of training data and n-gram order affect accuracy and realization times.

3.2 CCG Chart Realization

A high-level description of the chart realization algorithm follows.

The input to the algorithm is a logical form (LF), together with a function that computes n-gram scores of possible realizations. Note that n-gram scoring is integrated into the realizer's search algorithm, rather than being used in a post-process to rank alternative realizations, as has been more typically the case.

In the first phase of the algorithm, for each elementary predication in the input LF, lexical entries indexed by the predicate are accessed and instantiated, and the resulting edges are added to the (initially empty) agenda; also, based on this set of instantiated edges, potentially relevant semantically null entries are identified. In the second, main phase of the algorithm, edges are successively moved from the agenda to the chart and combined—using all applicable combinatory rules—with the edges already on the chart (or with the potentially relevant semantically null edges), with any resulting new edges added to the agenda, until no more combinations are possible and the agenda becomes empty, or until the time limit for the anytime search is exceeded.[2]

[2] In addition to the overall time limit, the implementation also supports a *new best* time limit, which caps the amount of time spent looking for a better scoring realization than the first complete one.

Table 1 Test suite sizes.

	LF/target pairs	*Unique up to SC*	*Length*			*Input nodes*		
			Mean	*Min*	*Max*	*Mean*	*Min*	*Max*
COMIC	549	219	13.1	6	34	8.4	2	20
Worldcup	276	138	9.2	4	18	6.8	3	13

The basic algorithm can be straightforwardly adapted to perform best-first anytime search, by simply treating the agenda as a priority queue sorted by n-gram scores. Since the search method is independent of how edges are combined via CCG's combinatory rules, we expect it to be applicable to chart realization algorithms for other grammatical frameworks as well.

The n-gram scores may also be used in n-best edge pruning, which limits the number of edges in the chart that can have equivalent categories (but different strings), removing the edge whose string has the lowest n-gram score when the n-best limit is exceeded. In addition to anytime search and n-best edge pruning, the OpenCCG implementation also employs four other efficiency methods—index filtering, LF chunking, feature-based licensing and instantiation of edges, and caching of category combinations—the first two of which are essential for adequate performance [18, 19].

3.3 Case Study

To assess the extent to which n-gram scoring can guide the anytime chart realization algorithm towards preferred realizations and reduce realization times, we measured the realizer's accuracy and speed, under a variety of configurations, on test suites for two small but linguistically rich grammars:

COMIC The COMIC grammar partially implements Steedman's [16] theory of information structure and prosody in CCG, and the core of the grammar is shared with the one deployed in the FLIGHTS system [12]. As shown in Table 1, the test suite contains 549 unique pairs of logical forms and target sentences, out of which 219 are unique after replacing certain words with semantic classes (e.g., replacing *Armonie* by *SERIES*). The test suite was derived by running the system through a range of simulated dialogues; deduplicating the generated logical forms; realizing the logical forms using a language model derived from a smaller regression test suite for the grammar; and manually correcting the resulting realizations to obtain the desired target sentences. The target sentences average 13.1 words in length, with a minimum of 6 and a maximum of 34 words. In these sentences, pitch accents such as H* and L+H* are considered integral parts of words, whereas boundary tones such as LH% and LL% are treated as separate words, like punctuation marks. The input logical forms range from 2 to 20 nodes and have 8.4 nodes on average.[3] An example sentence is *once_again L+H* LH% there are floral_H* motifs_H* LH% and geometric_H* shapes_H* on the decorative_H* tiles LL% , but L here L+H* LH% the colours are off_white_H* LH% and dark_red_H* LL%* .

Worldcup The worldcup grammar is from a linguistic study of extraction and coordination, and covers heavy NP shift, non-peripheral extraction, parasitic gaps, particle shift, relativization, right node raising, topicalization, and argument cluster coordination. The test suite contains five additional invented variants for each of the 46 pre-existing phrases discussed in [3], for a total of 276 unique pairs of logical forms and target phrases, half of which are unique after semantic class replacement. The phrases average 9.2 words in length, and vary from a minimum of 4 words to a maximum of 18 words. The number of nodes in the input logical forms averages 6.8, and ranges from 3 to 13. Example phrases include *game that John watched without enjoying* and *John knew that Brazil would defeat and Bill predicted that China would tie with Turkey.*

While these two grammars use unification in the usual way to handle phenomena such as person, number and case agreement, they have both been allowed to overgenerate to varying extents, in order to streamline grammar development. In particular, neither grammar sufficiently constrains modifier order, which in the case of adverb placement especially can lead to a large number of possible orderings. Additionally, the COMIC grammar allows for a one to many mapping from

[3] The number of nodes essentially corresponds to the number of content words.

Table 2 Accuracy measures and realization times (in ms.) for different n-gram scoring methods, with the COMIC test suite and 3-best pruning.

			Time 'til First		*Time 'til Best*		*Time 'til All*	
	Exact	*Score*	*Mean (±σ)*	*Max*	*Mean (±σ)*	*Max*	*Mean (±σ)*	*Max*
Baseline 1	241/549	0.75	459 (±323)	1564	459 (±323)	1564	483 (±328)	1575
Baseline 2	41/549	0.38	371 (±239)	1490	371 (±239)	1490	479 (±321)	1560
Topline	549/549	1	148 (±90)	749	150 (±92)	749	517 (±353)	1743
N6/cv25	**548/549**	0.99	**196 (±134)**	995	**196 (±134)**	995	532 (±381)	1782
WB6/cv25	548/549	0.99	251 (±177)	1360	251 (±177)	1360	530 (±379)	1768
MLE6/cv25	503/549	0.97	201 (±178)	1387	201 (±178)	1387	533 (±378)	1778
Azul6/cv25	539/549	0.99	174 (±110)	761	177 (±113)	761	558 (±410)	1960

themes or rhemes to boundary tones, yielding many variants that differ only in boundary tone type or placement. This flexibility makes it possible to handle discontinuous themes or rhemes, but it does so at the expense of making the grammar considerably more challenging for the realizer to process efficiently.

Using these two test suites, we timed how long it took on a 2.2 GHz Linux PC to realize each logical form under each realizer configuration.[4] To measure accuracy, we counted the number of times the best scoring realization exactly matched the target, and also computed a modified version[5] of the Bleu n-gram precision metric [13] employed in machine translation evaluation, using 1- to 4-grams, with the longer n-grams given more weight.

To rank candidate realizations, we used n-gram backoff models of orders 2 through 6, with semantic class replacement, created using the SRI language modeling toolkit [17] in a cross-validation setup. Note that since the test suite was derived from the output of the generator, it contains many more repeated phrases than one finds in a corpus of human dialogues, and thus data sparsity is not as large an issue as it is in speech recognition. We compared three different discounting methods: Ristad's natural (code: N) discounting law [14]; Witten-Bell (code: WB) discounting [20]; and no discounting (maximum likelihood estimation, code: MLE).[6] We also tried using our modified Bleu score (code: Azul), computing n-gram precision against all training examples. All of the n-gram scorers included an *a/an*-filter, which assigns a score of zero to sequences containing *a* followed by a vowel, or *an* followed by a consonant.

To gauge how the amount of training data affects performance, we ran a series of cross-validation tests, with 25 as the maximum number of folds. Since performance turned out to be already quite good with 2-fold cross validation, we also included a number of 1.*x*-fold tests, where the second fold is 0.*x* as large as the first, and both folds use the same amount of training data. For example, with 1.5-fold cross-validation, the first fold contains two-thirds of the data as test cases, while the second fold (half as big) has the remaining one-third of the data as test cases, and both folds use one-third of the data for training.

Finally, we compared the realization results using the n-gram scorers described above with two baselines and one topline (oracle method). The first baseline assigns all strings a uniform score of zero, and adds new edges to the end of the agenda, corresponding to breadth-first search. The second baseline uses the same scorer, but adds new edges at the front of the agenda, corresponding to depth-first search. The topline uses the modified Bleu score, computing n-gram precision against just the target string, a technique which we have found to be very useful for regression testing the grammar; clearly though, the topline represents an unrealistic scenario for applications, since if we already knew the target string, there would be no point in generating it.

3.4 Results and Discussion

Tables 2 and 3 show the accuracy measures and realization times for the COMIC and Worldcup test suites, respectively, for the 25-fold cross-validation case, where the scoring methods used n-grams

[4] Running the tests under different Linux and Windows Java virtual machines did not appear to change the relative timings.

[5] Our version did not include the bells and whistles intended to make cheating the Bleu metric more difficult. Also, the individual n-gram scores were combined using rank-order centroid weights, rather than the geometric mean, so as to avoid problems with precision scores of zero, when used with short phrases.

[6] We used an open vocabulary and no min counts. In earlier tests, we found that the default method—Good-Turing—generated a large number of warnings due to the small size of the data set, so we did not include this method here.

Table 3 Accuracy measures and realization times (in ms.) for different n-gram scoring methods, with the Worldcup test suite and 3-best pruning.

			Time 'til First		*Time 'til Best*		*Time 'til All*	
	Exact	*Score*	*Mean (±σ)*	*Max*	*Mean (±σ)*	*Max*	*Mean (±σ)*	*Max*
Baseline 1	86/276	0.6	152 (±182)	1222	152 (±182)	1222	175 (±216)	1342
Baseline 2	70/276	0.54	115 (±144)	889	115 (±144)	889	177 (±218)	1353
Topline	276/276	1	49 (±34)	181	50 (±34)	181	187 (±234)	1469
N6/cv25	**252/276**	0.94	**87 (±61)**	371	**87 (±61)**	371	195 (±249)	1594
WB6/cv25	254/276	0.94	100 (±89)	724	100 (±89)	724	195 (±249)	1590
MLE6/cv25	256/276	0.94	79 (±52)	275	79 (±52)	275	195 (±249)	1595
Azul6/cv25	260/276	0.96	68 (±57)	322	71 (±58)	322	203 (±260)	1669

of order 6. The realizer was configured to run with all efficiency methods turned on, including an n-best pruning limit of 3 edges per equivalent category, since 3 was the smallest value that allowed the topline method to achieve perfect accuracy. All test cases were allowed to run to completion, so that we could compare the times until the first and/or best realization was found to the times until all realizations were found.

The best performing n-gram model overall used Ristad's natural discounting. With both test suites, the N6 scorer achieved much higher accuracy than either baseline. With the COMIC test suite, the N6 scorer succeeded in ranking the target realization as the best one in all but one case—*there is also_H* artwork on the decorative tiles LL%*—where it mistakenly preferred *also_H** fronted, due to the trigram *there is also_H** appearing in just this example. With the Worldcup test suite, the N6 scorer did less well in ranking the target realization best, achieving exact matches in only 252 out of 276 cases. However, with the exception of a couple of topicalization choices,[7] the 24 mismatches appear to represent cases of acceptable free variation—e.g., differences involving the optional complementizer *that*, or alternative but acceptable placements of certain adverbs. Moreover, the rankings succeeded in avoiding many dispreferred variants allowed by the mildly overgenerating grammar, such as **easily Brazil defeated Germany*, **Marcos picked up it*, and *?Brazil easily defeated the team that China beat yesterday* instead of the target *Brazil easily defeated yesterday the team that China beat* (with *yesterday* modifying *defeated*).

While the WB6, MLE6 and Azul6 models achieved comparable accuracy with the Worldcup test suite, the MLE6 and Azul6 methods fared somewhat worse on the COMIC test suite. Since the MLE6 model uses no smoothing, it cannot back off to lower order n-grams when it encounters unseen sequences, which still occur even with 25 cross-validation folds; consequently, possible realizations receive probability scores of zero in such cases, as in the baseline models, with no remaining capacity to recognize dispreferred variants. With the Azul6 method, the n-gram precision weighting scheme gives such a strong preference to the longer observed n-grams that adverbs such as *also*, with medial position preferred, can get pushed to the side (i.e., initial or final position) when the medial realization involves an unseen n-gram sequence.

In regard to realization times, the N6 scorer yielded realization times that were better than WB6, most likely because it reserves less probability mass for unseen events. The N6 scorer's realization times were also considerably better than either baseline. For example, with the COMIC test suite, the N6 scorer found the first complete realization in 196 ms. on average, more than 2.5 times faster than the first baseline. What was somewhat surprising to observe was that the best scoring realizations nearly always appeared first, or soon after. With the N6 scorer, the first complete realization turned out to also be the best scoring one in all cases with the COMIC test suite, and in all but 8 cases with the Worldcup test suite, with a neglible effect on the average time. In contrast, the average time until all realizations were found was much higher, with greater variance, and with many cases taking more than a second. In the context of this comparison, it is important to note that with higher n-best pruning values, the differences become more dramatic. For example, with 5-best pruning and the COMIC test suite, the maximum time to find all realizations goes up to 2.5 seconds, while the times to find the first or best realizations remain essentially the same.

Turning now to how the amount of training data affects performance, Figure 1 shows that the times until the first complete realization is found decrease as the number of folds increases, for all four

[7] With the Worldcup grammar, topicalization has no semantic reflex in the logical form; in contrast, with the COMIC grammar, topicalization choices in the realizer are determined by a feature in the input logical form, rather than being left for the n-grams to try to decide.

Figure 1 Mean time (in ms.) until first realization is found using different n-gram scoring methods and n-grams of order 6, for cross-validation tests with increasing numbers of folds.

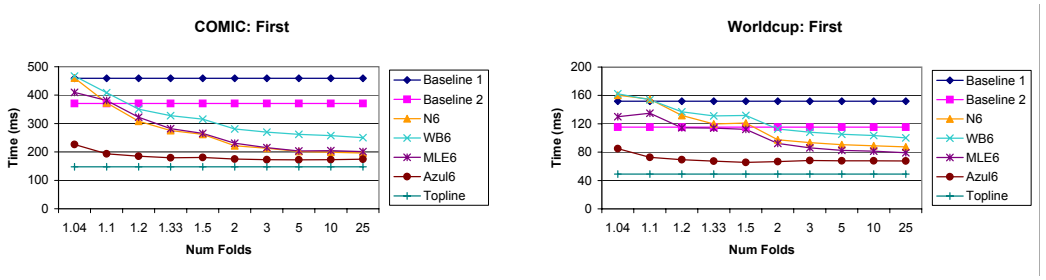
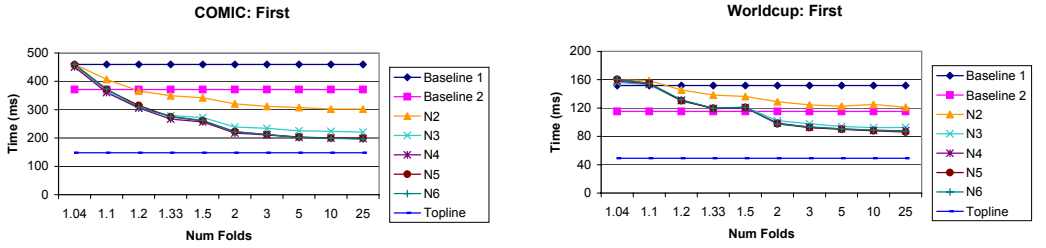


Figure 2 Mean time (in ms.) until first realization is found using n-grams of different orders and Ristad's natural discounting, for cross-validation tests with increasing numbers of folds.



n-gram scorers, with a relatively smooth progression from the first baseline towards the topline. Not surprisingly, the n-gram precision scorer (Azul6) shows less sensitivity to the amount of training data, but as discussed above, this method fares slightly worse on accuracy than the scorers using natural and Witten-Bell discounting.

To examine the effect of different n-gram orders, Figures 2-4 show, for Ristad's natural discounting method, how realization times decrease and accuracy increases when longer n-grams are employed. Figure 2 shows that trigrams offer a substantial speedup over bigrams, while n-grams of orders 4-6 offer a small further improvement. Figures 3 and 4 show that with the COMIC test suite, all n-gram orders work well, while with the Worldcup test suite, n-grams of orders 3-6 offer some improvement over bigrams.

A final question concerns the correlation between perplexity and realizer performance. Table 4 shows, for the two test suites, the Pearson correlation coefficients between perplexity and the number of exact matches, and between perplexity and the times until the first complete realization is found. As might be expected, perplexity correlates reasonably well with both measures when the MLE pairs are excluded, since the MLE perplexities are artificially low due to cases involving zero probabilities. Whether perplexity correlates well enough to predict whether one n-gram scorer will yield better realizer performance than another, however, needs further investigation.

3.5 Related and Future Work

Our work on using n-gram scoring to select preferred realizations follows the line of research pioneered by Knight and Hatzivassiloglou [9], and further investigated in e.g. [10, 4, 11]. It differs from this work, however, in its use of state-of-the-art grammars and semantics, with only mild overgeneration; though small and manually crafted, the current grammars have proved to be adequate for use in the COMIC and FLIGHTS dialogue systems, which operate in limited domains. In contrast, most of the work following [9] has employed wide coverage but massively overgenerating grammars. Consequently, these prior approaches have been unable to achieve very high quality, which is more important in dialogue systems than wide coverage.

Another difference is that our approach currently leaves very little lexical choice to the realizer. In

Figure 3 Number of realizations exactly matching target using n-grams of different orders and Ristad's natural discounting, for cross-validation tests with increasing numbers of folds.

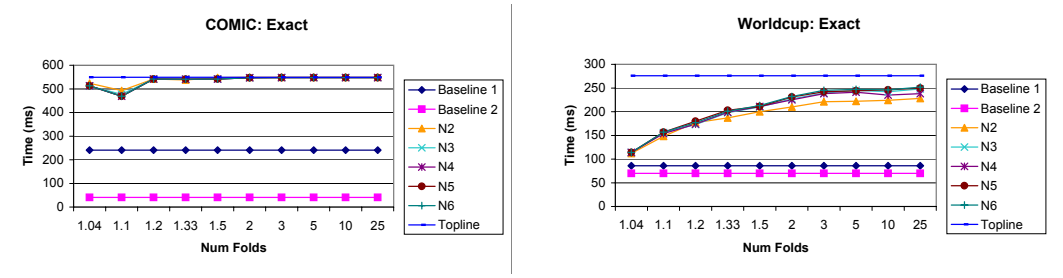
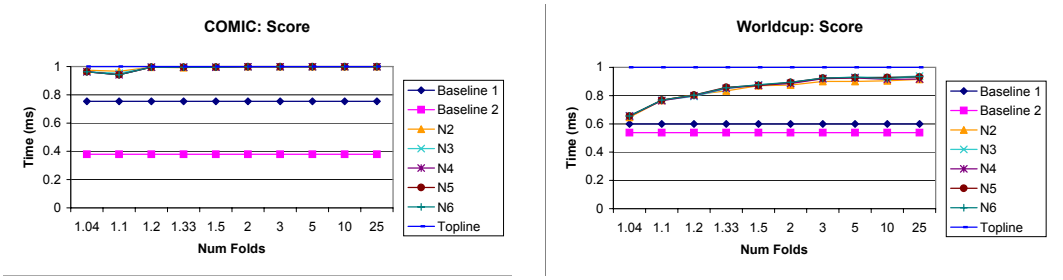


Figure 4 Modified BLEU scores using n-grams of different orders and Ristad's natural discounting, for cross-validation tests with increasing numbers of folds.



future work, we plan to investigate allowing the input logical forms to underspecify lexical choice in a flexible way. Whether our approach will continue to work equally well when faced with more underspecified input logical forms, however, remains to be seen.

The way in which we integrate n-gram scoring of possible realizations into the chart realization algorithm also differs from the above-cited work, where packed representations of a potentially exponential number of possible realizations are typically built in a first stage, and then in a second stage, a search is performed for high scoring realizations enumerated from the packed structure. In both approaches, a number of pruning strategies may be employed; since the OpenCCG realizer can take a function implementing a specific pruning strategy as an optional parameter, we assume that the two-stage approach and the anytime approach with integrated n-gram scoring can be put on equal footing as far as pruning goes. We may observe, though, that the two-stage approach typically (i) requires completing the packed chart; (ii) imposes an inflexible ordering on the enumeration and scoring of possible realizations; and (iii) does not terminate until the unpacking phase finishes. In contrast, the anytime approach employs a flexible search order, and does not require completing the chart. However, it does introduce two inefficiencies from the perspective of enumerating all realizations: (i) it introduces the overhead of a hash map in order to avoid combining equivalent categories multiple times, which is not necessary when using a packed representation; and (ii) edges may be introduced into the chart which are only later pruned when a better scoring edge with the same category emerges. If these inefficiencies are small compared to the time gains obtained, then we may expect the anytime approach to be better suited to the responsiveness requirements of dialogue systems; determining whether this holds in practice is a topic for future research.

4 Measuring the Effect of Realization Choices on Synthesis Quality

4.1 Overview

In spoken language dialogue systems, natural language generation and speech synthesis have often been handled completely separately. More recently, several researchers have begun to experiment with augmenting the text output of a generator with markup intended to improve the quality of the

Table 4 Pearson coefficients for correlations between perplexity and number of exact matches or times until first realization.

	Exact		*Time 'til First*	
	N,WB,MLE	*N,WB*	*N,WB,MLE*	*N,WB*
COMIC	-0.4	-0.51	0.68	0.88
Worldcup	-0.38	-0.95	0.18	0.92

synthesized speech. However, work in this vein has rarely considered—in any systematic way—how generation choices are likely to affect the quality of the speech synthesized from the resulting text. Moreover, while recent advances in speech synthesis using unit selection techniques have led to much improved overall quality, there usually remains a large degree of variability between what can be synthesized well and what comes out quite poorly.

A recent exception to the largely separate treatment of generation and synthesis is Bulyko and Ostendorf's work [6], where they make an initial attempt to integrate and jointly optimize generation and synthesis. In their approach, they take advantage of the fact that there are usually multiple acceptable ways to respond in a given dialogue state. Taking advantage of this flexibility, they modified a template-based generator to produce multiple wording and prosodic realizations of a target utterance, so that the synthesizer could choose an option that was likely to be synthesized well—e.g., by avoiding poor quality joins arising from inconsistent coverage in the unit database. With this method, they were able to show through perception experiments that coupling generation and synthesis can yield higher quality speech than the usual sequential implementation.

While Bulyko and Ostendorf's method is elegant, it requires a synthesizer that is prepared to search through a graph of input possibilities, which is not how most synthesizers currently work. Thus, as an alternative approach to loosely coupling generation and synthesis, we suggest using n-grams from the recording script used to build the synthetic voice to guide realization choices, to improve the chances that a particular realization can be synthesized using large stretches of units from the same recorded utterance.

To assess the potential of this approach, we carried out a perception experiment, to appear in Rocha's forthcoming MSc thesis [15], examining the extent to which realizer choices—guided by such n-grams—can affect the perceived quality of synthesized speech. The details and results of this experiment are described in the remainder of this section.

4.2 Limited Domain Unit Selection Synthesis with APML

In unit selection synthesis, the goal is to concatenate pre-recorded segments of speech in a way that both covers the input string and sounds as natural as possible. Typically, a target utterance structure is predicted and suitable candidates from the inventory are proposed for each target unit; the best candidate sequence is then found by minimising target and join costs, using a Viterbi search. There are a number of options for what size these units can be, the main contenders being phones, half phones, diphones or larger units; and likewise, there are many ways of restricting the set of target units and setting target costs. The particular choices one employs affect flexibility, quality, and search times.

The voice for the perception experiment uses the technique described in Baker's MSc thesis [2, 1], with sentences in the FLIGHTS [12] domain. In this approach, which extends Black and Lenzo's [5] cluster unit synthesis approach to limited domain synthesis, unit candidates are restricted to ones with the same phone and appearing in the same word, with the same pitch accent and with the same following boundary tone (or lack thereof). Pitch accents and boundary tones are specified in the synthesizer input using APML. By limiting unit candidates in this fashion, it is possible to produce contextually appropriate intonation, but the approach suffers from lack of flexibility, insofar as it requires the range of sentences to be synthesized to be precisely anticipated at the time of constructing the recording script. To address this problem, work is currently underway to extend Festival 2 [8] to handle APML input in a more flexible way.

4.3 Experimental Design

A forced-choice experiment was designed to test the hypothesis that the participants would prefer the synthesized sentences generated with the use of n-gram scoring.

In this experiment, ten sentences were presented in pairs to the participants. They were asked to choose the version that sounded better. They could listen to the sentences as many times as they wanted to before making their choice. After making their choice, however, they could not go back and change their option. Half of the sentences with n-gram scoring were presented first, and in the other half, the sentences with no n-gram scoring. Additionally, the order of the sentence pairs was randomized. These procedures were adopted in order to control for order effects.

The test sentences were presented to the subjects via Sennheiser headphones with a comfortable level of hearing using standard audio software. E-prime was the software used to present the sentences on the computer.

4.4 Test Sentences

The ten test sentences were taken from the examples discussed [12]. The following ten sentences were exactly realized using n-grams derived from these target sentences alone (where pitch accents are considered integral parts of words, and boundary tones are treated as separate words):

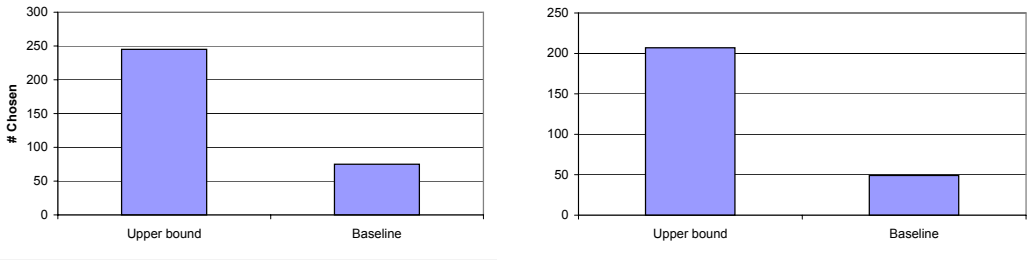
1. There's a direct_H* flight on BMI_H* with a good_H* price_H* LL%.
2. It arrives at four_ten_pm_H* LH% and costs one_hundred_H* and twelve_pounds_H* LH%.
3. The cheapest_L+H* flight LH% is on Ryanair_H* LL%.
4. It arrives at twelve_forty_five_pm_H* LH% and costs just fifty_pounds_H* LL%, but it requires a connection_H* in Dublin_H* LL%.
5. There's a KLM_H* flight LL% arriving Brussels_H* at four_fifty_pm_H* LL%, but business_class_H* is not_H* available_H* LH% and you'd need to connect_H* in Amsterdam_H* LL%.
6. If you want to fly direct_L+H* LH%, there's a BMI_H* flight LL% that arrives at four_ten_pm_H* LL%, but it has no_H* availability_H* in business_class either LL%.
7. There are_L+H* seats in business_class LH% on the British_Airways_H* flight LL% that arrives at four_twenty_pm_H* LL%.
8. It requires a connection_H* in Manchester_H* though LL%.
9. You can fly business_class_H* on British_Airways_H* LL%, arriving at four_twenty_pm_H* LL%, but you'd need to connect_H* in Manchester_H* LL%.
10. There is a direct_L+H* flight LH% on BMI_H* LL%, arriving at four_ten_pm_H* LL%, but it has no_H* availability_H* in business_class LL%.

As discussed in section 3, using n-grams from the target sentences establishes an upper bound on expected performance; however, since our cross-validation experiment showed that near perfect accuracy can be achieved in the COMIC domain, we have reason to believe that realization results in actual practice will be close to this upper bound.

For the experiment, the same logical forms were realized with no n-gram scoring, with the first complete realization licensed by the grammar used as a baseline reference:

1. There is **an** BMI_H* direct_H* flight with **an** good_H* price_H* LL%.
2. It arrives at four_ten_pm_H* LH% and costs one_hundred_H* and twelve_pounds_H* LH%.
3. The cheapest_L+H* flight LH% is on Ryanair_H* LL%.
4. It arrives at twelve_forty_five_pm_H* LH% and costs just fifty_pounds_H* but it requires **an** connection_H* in Dublin_H* LL%.

Figure 5 Number of times synthesized speech from upper bound realizations vs. baseline ones were chosen as better, across all ten sentences (left), and excluding the two identical ones (right).



5. There is **an** KLM_H* flight arriving Brussels_H* at four_fifty_pm_H* but business_class_H* is not_H* available_H* LH% and you'd need to connect_H* in Amsterdam_H* LL%.
6. If you want to fly direct_L+H*, LH% there is **an** BMI_H* flight that **arrive** at four_ten_pm_H* but it has no_H* availability_H* in business_class either LL%.
7. There're_L+H* seats in business_class LH% on the British_Airways_H* flight LL% that **arrive** at four_twenty_pm_H* LL%.
8. It though requires **an** connection_H* in Manchester_H* LL%.
9. You can fly business_class_H* on British_Airways_H***,** arriving at four_twenty_pm_H* but you'd need to connect_H* in Manchester_H* LL%.
10. There is **an** direct_L+H* flight LH% on BMI_H* LL%, arriving at four_ten_pm_H* LL% but it has no_H* availability_H* in business_class LL%.

Note that in the absence of n-gram scoring, the order in which realizations appear is unpredictable, so the baseline realization may be considered to be one randomly chosen from those licensed by the mildly overgenerating grammar. In particular, note that sentences 2 and 3 turned out to be identical, while the other sentences exhibited a range of more or less subtle differences, including the choice of *a/an*, the presence of a comma for pausing and the inclusion of some boundary tones, whether or not the copula was contracted, number agreement on *arrive* in relative clauses, and the position of adjectival or adverbial modifiers such as *direct* or *though*.

Since some changes were made to the APML used to build the limited domain voice during Rocha's project—to better match what the speaker actually said in the recording sessions—it was necessary to modify some of the test sentences in order to find matching combinations of words, pitch accents and boundary tones in the unit database. In particular, the H* pitch accent on *price* had to be removed, and some LH% boundaries had to be changed to LL% ones. However, as the modifications were made in both versions, they should have no effect on the results of the experiment.

4.5 Subjects

The participants of this experiment were 32 native-speakers of English without any known hearing or language deficit. The subjects also participated in an earlier experiment comparing dialogue turns, as described in [15]. Both experiments together lasted about 25-30 minutes and subjects were paid £5 at the completion of them.

4.6 Results

Across all subjects and all ten sentence pairs, the synthesized speech from the upper bound realizations were preferred in 245 cases out of 320, which represents 77% of the total. If subjects had no preference, we would expect the upper bound versions to be chosen only half the time, i.e. in only 160 cases. Using a binomial test, we can reject the null hypothesis of no preference with very high confidence ($p < 0.0001$). As expected, excluding sentences 2-3, which were identical, yields

an even higher percentage of 81%, as the upper bound realizations were preferred in 207 out of 256 remaining cases. Figure 5 shows the relative frequencies of the subjects' preferences side-by-side.

5 Conclusions

In this report, we have described the experiments with multimodal output undertaken in the COMIC project to date, emphasizing the two aspects of multimodal output that have not been examined in other deliverables, namely instance-based English surface realization and its connection with speech synthesis. In particular, we reported on (1) cross-validation experiments showing how n-grams from a training corpus can be used to guide the realizer to quickly produce target realizations with high accuracy, and (2) a perception experiment showing that the resulting realizations positively affect the perceived quality of synthesized speech produced from them, when compared to baseline realizations, with very high significance.

In the current COMIC system, we have not made use of the prosodic specifications produced by the realizer, as the limited domain unit selection voices that accept APML input have proved too brittle for ongoing use. However, before the end of the project, we plan to take advantage of work that is in progress to extend Festival 2 [8] to handle APML input with a unit selection voice in a more flexible way. We also plan to extend the surface realizer's n-grams to choose among multimodal n-grams, where information about the gestures and facial expressions accompanying the words is incorporated into the search for high-scoring surface forms.

References

References

[1] Rachel Baker, Robert A. J. Clark, and Michael White. Synthesising contextually appropriate intonation in limited domains. In *Proc. of the 5th ISCA Speech Synthesis Workshop*, 2004.

[2] Rachel Elizabeth Baker. Using unit selection to synthesise contextually appropriate intonation in limited domain synthesis. Master's thesis, Department of Linguistics, University of Edinburgh, 2003.

[3] Jason Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, School of Informatics, University of Edinburgh, 2002.

[4] Srinivas Bangalore and Owen Rambow. Exploiting a probabilistic hierarchical model for generation. In *Proc. COLING*, 2000.

[5] Alan Black and Kevin Lenzo. Limited domain synthesis. In *Proc. ICSLP-2000*, 2000.

[6] Ivan Bulyko and Mari Ostendorf. Efficient integrated response generation from multiple targets using weighted finite state transducers. *Computer Speech and Language*, 16(3–4):533–550, 2002.

[7] B. De Carolis, C. Pelachaud, I. Poggi, and M. Steedman. APML, a mark-up language for believable behavior generation. In H. Prendinger and M. Ishizuka, editors, *Life-like Characters: Tools, Affective Functions and Applications*. Springer, 2004. in press.

[8] Robert A. J. Clark, Korin Richmond, and Simon King. Festival 2 — build your own general purpose unit selection speech synthesiser. In *Proc. of the 5th ISCA Speech Synthesis Workshop*, 2004.

[9] Kevin Knight and Vasileios Hatzivassiloglou. Two-level, many-paths generation. In *Proc. ACL*, 1995.

[10] Irene Langkilde. Forest-based statistical sentence generation. In *Proc. NAACL*, 2000.

[11] Irene Langkilde-Geary. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. of the Second International Natural Language Generation Conference*, 2002.

[12] Johanna Moore, Mary Ellen Foster, Oliver Lemon, and Michael White. Generating tailored, comparative descriptions in spoken dialogue. In *Proceedings of FLAIRS-04*, 2004.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176, IBM, 2001.

[14] E. S. Ristad. A Natural Law of Succession. Technical Report CS-TR-495-95, Princeton Univ., 1995.

[15] Neide Franca Rocha. Evaluating prosodic markup in a spoken dialogue system. Master's thesis, Department of Linguistics, University of Edinburgh, 2004.

[16] Mark Steedman. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689, 2000.

[17] Andreas Stolcke. SRILM — An extensible language modeling toolkit. In *Proceedings of ICSLP-02*, 2002.

[18] Michael White. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language and Computation*, 2004. To appear.

[19] Michael White. Reining in CCG chart realization. In *Proc. of INLG-04*, 2004.

[20] I. H. Witten and T. C. Bell. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Trans. Information Theory*, 37(4):1085–1094, 1991.